Regioned Downsample for ANN Image Classification

Alternate Selection Methods

Keith A Greenhow
School of Computing
University of Kent
Canterbury, Kent
Email: kg246@kent.ac.uk

Colin G Johnson
School of Computing
University of Kent
Canterbury, Kent
Email: C.G.Johnson@kent.ac.uk

***Abstract*—In an earlier paper, a novel method to pre-process image data for use in Artificial Neural-Network (ANN) classification was presented. This method requires an additional training stage prior to the main learning phase of the ANN. In this extra stage, an additional algorithm (a Selection method) is used to generate the data that is required to construct the final pre-processor. As part of the introduction of that method, it was presented with a single Selection method that was termed Saliency Heat Mapping. This paper will present a number of alternative Selection methods and compare how effective they are against a sample problem.**

Keywords*—*Artificial Neural Networks, Preprocessor, Image Processing, Salience, Relevance Assessment

I. Introduction

This paper present two new methods for preprocessing of visual data for use in Artificial Neural Network (ANN) based classification. This take the form of two new selection methods for the Regioned Downsample algorithm (Greenhow and Johnson, 2014) (from which this paper extends). First, this paper will providing an overview of the algorithm's functionality and the related terminology. Following this with an explanation of the previous selection method and the additional novel approaches. The comparison of the methods is accompanied by the structure of the comparison model (a simple naïve face detection problem) and results from these test. This is then followed up with the conclusion drawn from these results. Potential avenues for continued effort are the provided to end.

The overall question that is being asked is: what algorithm should be used to sample a large input (such as an image) to present a reasonable number of inputs to a neural network which will be used to carry out classification or prediction based on those inputs?

II. Overview

This section will cover the terminology used in this paper and the functionality of the Regioned Downsample Preprocessor.

A. Saliency

In this paper, the termed *Saliency* is used to describe how 'useful' each input is in generating predictions or classifications. From the earlier paper: "we are not interested in the contribution of the inputs to the ANN's predictive accuracy, but rather to the ANNs output function". When dealing with Artificial Neural-Networks (ANNs), the saliency of each unit can be computed (similarly to back-propagation) as

$$\hat{\rho}_i = \begin{cases} 1 & \text{when } i \text{ is an output,} \\ \tanh\left(\sum_j^{N_i} |w_{ij}|\, \hat{\rho}_j\right) & \text{otherwise.} \end{cases} \quad (1)$$

where, N_i is the set of subsequent neuronal units that have unit i as an input source; w_{ij} is the synaptic weight between units i and j; and $\hat{\rho}_x$ is the approximated saliency of unit x.

This method was based on the earlier work by Mozer and Smolensky (1989).

B. Preprocessor

1) Definition: This paper discusses image preprocessors as used by an artificial neural-network (ANN) backed system or process. In this specific context the definition of a preprocessor as 'a data preparation stage, processes or algorithm in which image data is normalised, colour corrected and/or resized before being presented to a subsequent ANN for task specific processing' is used.

2) Traditional Preprocessors: When constructing image processors using an ANN it is standard practice to implement some form of preprocessor to deal with the problem of dimensionality. These preprocessors are traditionally implemented as calls to the well known Bilinear or Bicubic interpolation algorithms[1], typically used for image resizing in graphics applications. These algorithms are have been around for many years and modern implementations are extremely well implemented for reduced computation time.

The issue with these algorithms is that they are designed for human vision and have a constant information density. Though they have shown sufficiency in the past, it is not well known how suitable these algorithms are for ANN, or if there are better methods for these situations.

[1]The works by Shibata and Utsunomiya (2011); Davies et al. (2010) show good examples of this

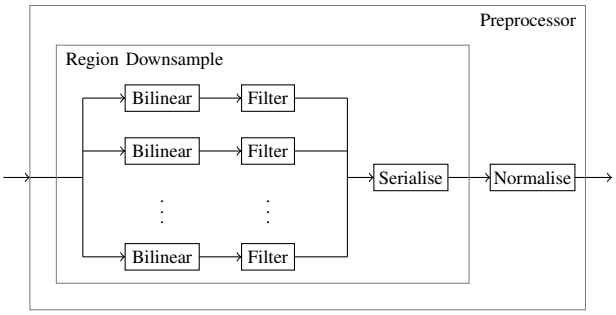


Fig. 1. Representation of the implementation of the Regioned Downsample Preprocessor, showing internal component topology and data flow.

3) Regioned Downsample Preprocessor: The implementation of the Region Downsample Preprocessor (RDP) is in the form of multiple small bilinear interpolation functions, each responsible for a small region of the input image and to output at a specified resolution (with possible overlapping regions). The multiple outputs are then filtered to only the intended pixels and then serialised to a single stream of colour pixels (See Fig. 1). The ordering of the serialisation *must* remain consistent between applications of the same preprocessor.

As part of constructing a RDP, first the pixels that need to be used need to be determined and the conversion of these selections to a compatible format. The selection process will be covered in detail in the next section so will be skipped as part of this overview. The conversion to a compatible format is a step referred to as *Optimisation*. At current, the Optimisation process currently uses a naïve optimisation which groups all selected pixels by their resolution and then defines a Bilinear interpolation to process the minimum region of all the pixels and a filter to remove the additional pixels.

III. Selection

Selection is the term used to describe the process of filtering out pixels at the multitude of resolutions until you have a minimal number of pixels that provide sufficient information to perform the intended task. The Saliency metrics described earlier are used in this process to approximate the quality of each pixel at the given resolution. The first of these Selection Methods is Saliency Heat Mapping. This is the method that was introduced in the previous paper. The two additional methods are Restricted Saliency Heat Mapping and Pruned-Bilinear Selection.

Preparation: Saliency Computation

Prior to selection, the saliency is precomputed to allow for use in the multiple competing selection methods. An upper and lower bound are defined, RES_{min} and RES_{max}, that mark the minimum and maximum resolutions to process (in both orthogonal directions). An ANN is generated with a traditional bilinear interpolation preprocessor and is trained using the Building set (see IV-C on how they are constructed). The Building set is the subset of the entire Training set that is solely used to *build* the ANN. After a fixed number of epochs, the ANN is parsed by the Saliency Metric and these saliency values are averaged with those from repeated runs at the same resolution. Each resolution's average saliency matrix

Fig. 2. Example matrix for an image from the Caltech-158 training set. Here RES_{min} and RES_{max} define the range of the matrix contents and are equal 8 and 30 respectively.

is paired up with its average accuracy (computed by running the ANN against the Validation set and scoring with Cohen's Kappa) and passed on the selection process. Additionally, to provide suitable comparison, the single resolution with the greatest accuracy according to the Validation set is used as the basis for comparison with the traditional methods (Bicubic and Bilinear interpolation). The Validation set is the remainder of the Training set (after exclusion of the Building set) used to validate that learning has been successful and provide some approximation of the quality of said learning.

A. Saliency Heat Map Selection

Saliency Heat Map Selection is the original Selection method that was implemented along with the Regioned Preprocessor in the earlier paper. This method takes the image matrices provided by the Saliency Computation stage and resize via nearest-neighbour interpolation, such that their dimensions are $2 \times RES_{max}$. The resultant matrices are then averaged together, weighted by their accuracies (from the Saliency Computation Stage), generating a single Saliency Heat Map (See Fig 3 for the SHM generated.). From here a approximate saliency can be generated for all pixels, from all resolutions in the $RES_{min} \ldots RES_{max}$ range by looking at the average value within the area covered by that pixel on the SHM. The single pixel with the highest approximated saliency is selected and the area on the SHM it covers is zeroed (blacked out). This process is repeated until there are no pixels that are above a predefined threshold. For the implementation, a selection threshold of 0.65 was found to be sufficient by informal experimentation.

B. Restricted Saliency Heat Map Selection

This new method works in the same manner as SHM Selection, but applies the additional criteria to pixel selection. Prior to selecting pixels by highest predicted saliency, the pixels are pre-filtered so that only pixels from resolutions that had positive accuracy scores from the Validation tests (i.e. The ANNs that were used to generate the saliencies managed to learn the task at hand, even to a minor degree) can be chosen.

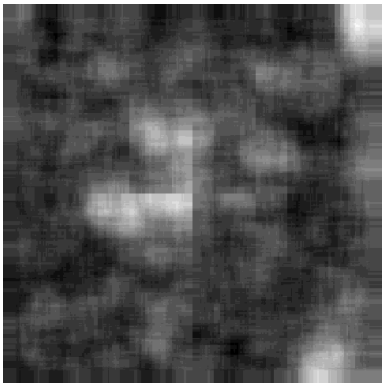


Fig. 3. The SHM generated for the naïve face finding task. The image originally ranged over the mid-greys (about 0.45 to 0.65) and has been normalised for easier viewing. From this with can see that (for this task) the left side of the profile, (right of centre in image above) has a higher saliency, implying greater predictive value.

C. Pruned-Bilinear Selection

This new method is named for the fact that it results in a selection that represents a standard bilinear interpolation preprocessor, but with some of its pixels ignored. This process uses the same process of selection as Restricted-SHM, but applies a much tighter constraint. Rather than only exploring resolutions that were even slightly successful in learning the task, it only selects from the single resolution that to perform the best on the Validation Set (see section III).

IV. Comparison

To compare the two newly proposed Selection methods against Saliency Heat Map Selection, a naïve face detection algorithm was implemented as test basis. The implementation of the main face detection system (an ANN) was left in a unoptimised state to best show the effects of the different preprocessors.

The method of comparing was split into two main tasks, Construction and Testing.

A. Construction

The process was broken down into three steps

1) The preparation stage is used to perform the Saliency Computations, as described in section III.
2) After preparation, the three sets of pixels are selected for by the three selection processes, using the recorded saliency measures. The selected pixels are then packaged into three preprocessors.
3) With preparation and Selection complete, training begins. Each of the three constructed preprocessors and an additional two control preprocessors (backed by a traditional Bicubic and Bilinear interpolation functions respectively) are each assigned to 200 randomly initialised ANNs forming a Preprocessor-ANN system[2]. Each system is then trained using Back-Propagation against the Building sets[3] (see section IV-C below for construction).

B. Testing

After training, each system used in a naïve face detection algorithm to attempt to find the faces in the Test set.

The naïve face detection algorithm is implemented as a sliding window over an image pyramid. The naïvety of this implementation allows for location and scale independence but does not consider orientation.

1) Each face detection system has a sliding window that is initially $^1/_4$ the size ($^1/_{16}$ the area) of the extracted faces from the Training set (see section IV-C3).
2) The sliding window is initialised to the top left corner.
   a) For each location of the sliding window, the test image is cropped to the window and then parsed to the systems preprocessor.
   b) The system then processes the prepared image data in the ANN and determines if the sliding window appears to be over a face (See section IV-D for detection specifics).
3) After processing the sliding window is shifted 20 pixels right and the process repeated.
4) At the end of each line, the sliding window is moved 20 pixels down and moved back to the left edge.
5) Once the sliding window has traversed the whole image, the size of the sliding window is increased by $^1/_3$ and moved back to the top left corner.
6) This is repeated until the sliding window covers the whole image.

C. Datasets

The following data sets where used for the naïve face detection system. (See Fig. 4 for samples.)

1) CMU-130: This data set was constructed from the union of the CMU frontal face data sets[4] A, B and C. The data set consists of numerous faces that are looking towards the camera in an upright manner, formatted in indexed grey-scale in a loss-less compression format. The data sets are provided with a ground truth labelling of each face in the scene, identifying the eyes, the nose and the corners and centre of the mouth.

2) CMU-Rotated: Equally, this data set comes from the CMU frontal face data sets; specifically the CMU Rotated Test Set[4]. This data set contains faces that are looking towards the camera, but are at slight angles from upright that can increase identification difficulty. These are provided with the same ground truth and data sets A-C.

3) Caltech-158: This data set is a subset of Caltech Faces 1999 (Front) data set[5]. The original data set includes 450 images of 27 unique faces that look towards the camera and are oriented upright. They are formatted in full RGB with negligible compression artefacts (due to the JPEG file format).

[2]The term *system* is used to describe each Preprocessor-ANN pair in a generic manner when the specific preprocessor is not known/identified or multiple types of preprocessor are being discussed.

[3]The Building set is used twice, once for initial Saliency computation and once for the training of the final ANNs.

[4]Data set A, C and the rotated data set were collected by Rowley et al. (1998), whilst data set B was compiled by Sung and Poggio (1998)

[5]Caltech Faces 1999 (Front) was compiled by Weber (1999)

(a) CMU-130 (b) CMU-Rotated (c) Caltech

Fig. 4. Three sample images from three data sets. These provide examples of a single image with multiple faces, an image with a single face proportionally small to the image, and an image with a face that fills most of the image.

The provided ground truth of this data set is incompatible with the implemented system, so the first 158 images were manually labelled to produce a data set slightly larger than the CMU data set A. The new ground truth was formatted in a consistent manner to that provided with the CMU data sets.

Subset construction: As data sets CMU-130 and Caltech-158 were to be used to construct the neural networks, they were segregated into the Build[6], Validation[6] and Test sets. To do this the images, along with associated ground-truths, of both sets were initially loaded into memory. The first 10% of faces were separated into the Test sets, T(CMU-130) and T(Caltech). The faces from the remaining images are extracted (cropped and scaled), along with an equal number of background samples[7]. The faces and background images are randomly assigned to either the respective Building set or their Validation set, so that 90% form the Building sets, B(CMU-130) and B(Caltech-158), whilst the remainder 10% form the Validation sets, V(CMU-130) and V(Caltech-158).

The data set CMU-Rotated is not used for training purposes and is used as an additional test set alongside T(CMU-130). This is specifically used to see if there is any noticeable increase or decrease in generalisation of the solution.

D. Implementation

The face detection algorithm was implemented as a standard feed-forward perceptron based neural network with a topology of $I \rightarrow 10 \rightarrow 2$ (where I represents the number of neurons required to fully represent the data provided by the accompanied preprocessor). Of the two output neurons, one was defined as a background identifying neuron and the other as a face detecting neuron. A face was deemed to have been identified by the ANN if and only if the face detecting neuron presented an activity of 0.95 or greater and had higher activity than the background identifying neuron. Learning was implemented using Back-Propagation and a learning rate of 0.8.

E. Results

To quantify the quality of each preprocessor, Percentage Correct Detection (CD) and Number of False Positives (FP) (Yang et al., 2000) are used to compare and contrast. A correct detection is recorded if the system classifies the current location of the sliding window to contain a face and the region covered by the sliding window contains the eyes, nose and mouth from the same face; otherwise it is identified as a false positive. To prevent erroneous scoring due to multiple detections of the same face, only the first detection is counted. Further detections of the same face are not counted towards CD or FP.

The graphs Fig. 5 and Fig 6 show the non-dominated result sets of the repeated trials (grouped by preprocessor). Note that not all averages are shown in these figures. For preprocessor systems in which this is he case, it is due to the average false positive rate being sufficiently high that if it where included, the important aspects of the graphs would me difficult to view.

In Fig. 5a, it can clearly be see that the systems that used a Pruned-Bilinear preprocessor dominate all of the other systems with generated/tested. It is also interesting to note that below 70% accuracy, The two traditional methods seem to have a false-positive rate about the same a Restricted-SHM Selection, which then increases significantly over that threshold.

Fig. 5b shows an expected outcome with regards to the non-dominated sets, in that they all appear to perform equally well when presented with slightly dis-similar inputs than what they were trained for. The averages show some interest properties in that the RDPs are all have higher correct detection rates on average than the traditional methods (significantly so for Pruned-Bilinear) at the expense of increased false positive rates.

Lastly, fig 6 shows almost the reverse outcome from the CMU trials. With the colour input images of the Caltech data set, the Pruned-Bilinear preprocessors had performed worse then the others due to a noticeably higher false positive rate. Restricted-SHM Selection proved the most effective at this task, dominating all others.

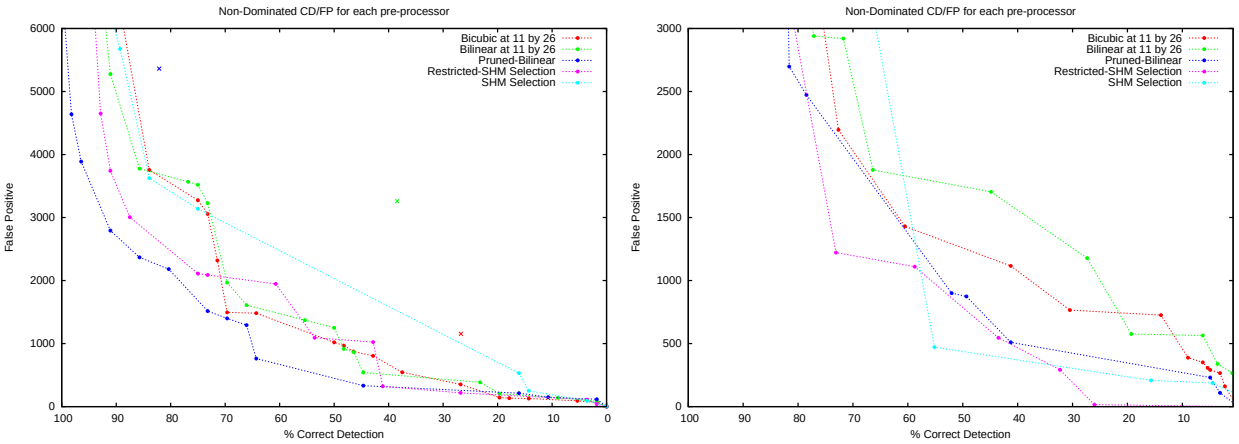
V. Conclusion

In conclusion, there was a significant improvement in the accuracy (in terms of false positive and correct detection rate) between the RDP and the traditional method, due mostly to the decrement in false-positive rates. Additionally there also appears to be a subtle difference between the different selection methods used to construct the RDP depending on the specifics of the task.

Against the grey-scale CMU images, the Pruned-Bilinear preprocessor performed best, where as Restricted-SHM generally performed the best when the system had to deal with the

[6]The Build and Validation sets collectively the make up the Training Set

[7]A background sample is considered to be a section of the image that would fail to register as a correct detection, as described in section IV-E.



(a) T(CMU-130) (b) CMU-Rotated

Fig. 5. The set of non-dominated systems, grouped by preprocessor and marked with circles, for verification against the T(CMU-130) data set (a). The crosses identify the *'average'* system for each given preprocessor group. Additionally the CMU-Rotated data set (b) (same layout) is used to explore any noticeable improvement or degradation in generalisation to the problem.

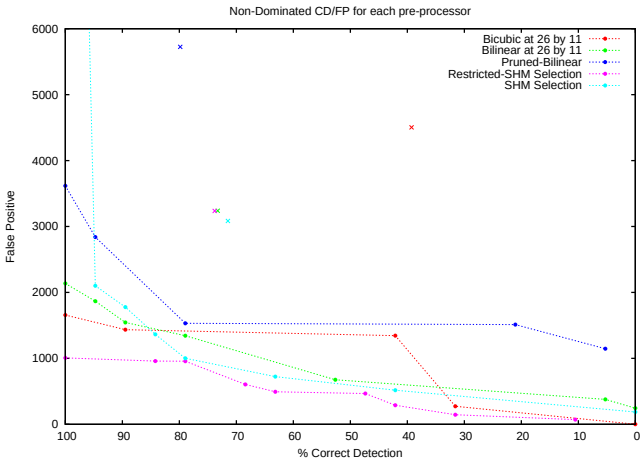


Fig. 6. The set of non-dominated systems, grouped by preprocessor and marked with circles, for verification against the T(Caltech-158) data set. The crosses identify the *'average'* system for each given preprocessor group.

rotated faces from CMU-Rotated. Against the RGB Caltech images, Pruned-Bilinear had a surprising poor performance in terms of false positive rates, with the traditional and SHM Selection methods performing mediocre and Restricted-SHM having the lowest false-positive rates.

VI. Future Work

A. Optimisation

In section II-B3, it was mentioned that a naïve Optimisation process was in use. The current implementation of this method is very simplistic. This is a prime location for improvement in the runtime of the preprocessor. Improvements to the optimiser will allow for the construction of RDPs that will run faster due to either a smaller number of bilinear interpolation preprocessors (reducing overheads), or by generating more numerous bilinear interpolation preprocessors that process smaller regions.

References

S. Davies, C. Patterson, F. Galluppi, A. Rast, D. Lester, and S. Furber, "Interfacing real-time spiking i/o with the spinnaker neuromimetic architecture," in *Proceedings of the 17th International Conference on Neural Information Processing: Australian Journal of Intelligent Information Processing Systems*, 2010, pp. 7–11.

K. A. Greenhow and C. G. Johnson, "Region based image preprocessor for feed-forward perceptron based systems," in *Advances in Neural Networks ISNN 2014*, ser. Lecture Notes in Computer Science, Z. Zeng, Y. Li, and I. King, Eds. Springer International Publishing, 2014, pp. 414–422. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-12436-0_46

M. C. Mozer and P. Smolensky, "Skeletonization: a technique for trimming the fat from a network via relevance assessment," *Advances in neural information processing systems*, vol. 1, pp. 107–115, 1989. [Online]. Available: http://dl.acm.org/citation.cfm?id=89851.89864

H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 1, pp. 23–38, Jan 1998.

K. Shibata and H. Utsunomiya, "Discovery of pattern meaning from delayed rewards by reinforcement learning with a recurrent neural network," in *The 2011 International Joint Conference on Neural Networks (IJCNN)*, Aug 2011, pp. 1445 –1452. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6033394&tag=1

K.-K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 1, pp. 39–51, Jan 1998.

M. Weber, "Caltech frontal face database," 1999.

M.-H. Yang, N. Abuja, and D. Kriegman, "Face detection using mixtures of linear subspaces," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, 2000, pp. 70–76.